

Categories

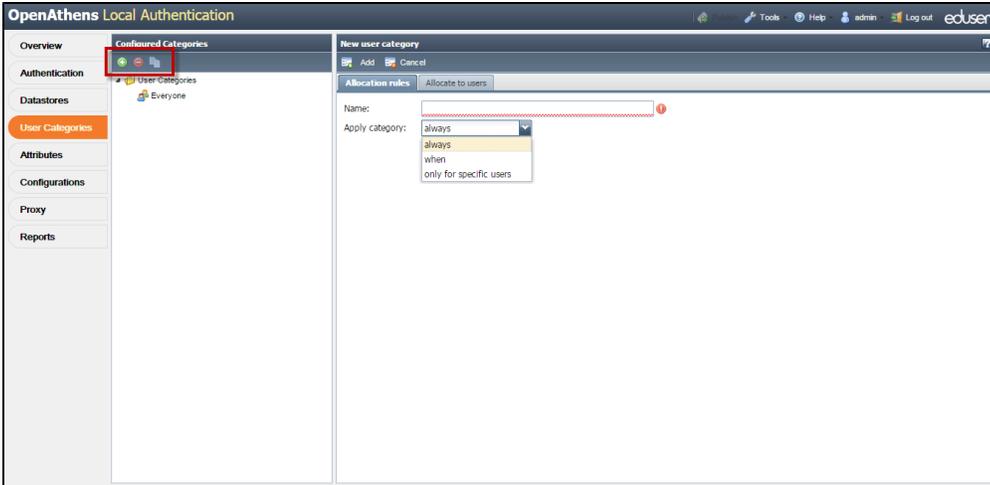
Categories are how you group your users and are used for two things in OpenAthens LA

- Assigning attributes
- Segmenting statistics

You should already have at least one category set up called Everyone. This is not everyone in the world of course, just everyone who is authenticated. You do need to keep this category as several functions depend on it.

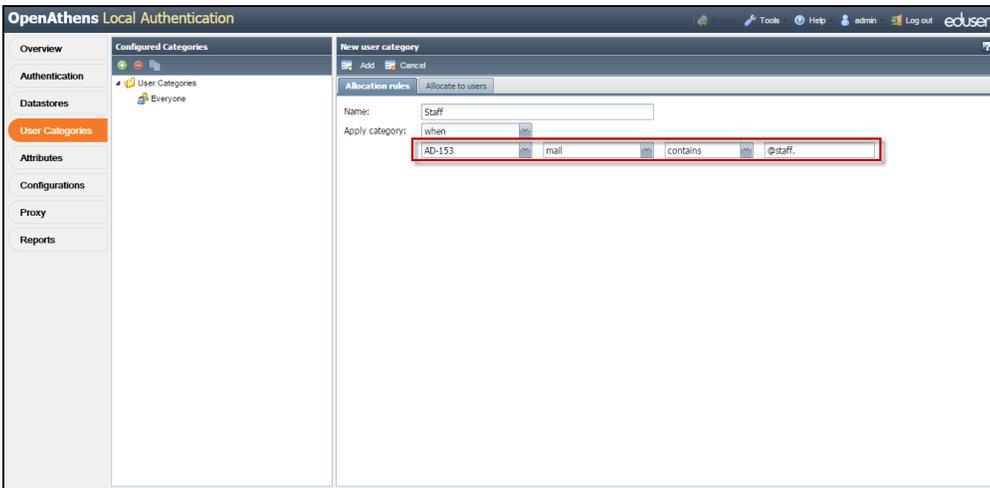
Creating categories

You can create a category based on any information in any connected data store. To create a new category, first click on the green plus button at the top of the list of existing categories.



The right hand panel will ask you for a category name (no spaces allowed) and when it will apply – ‘always’ or ‘when’. Selecting ‘when’ will allow you to specify a condition:

- From the first box select the data store to look at.
- The second box will now have a list of the fields that are available in that data store. Pick the one you need.
- The third box is the condition – e.g. matches, contains, ends with, etc.
- The fourth and final box is the value to go with the condition.



Once you've made your selection, click on the 'Add' button at the top of the panel. The runtime won't know about the new category until you publish, but you do not need to publish until you are ready for your changes to take effect.

Deleting categories

To delete a category, select it in the list on the left then click on the red minus button at the top of the list.

Other operations

To copy a category with a complicated setup you want to base a new category on, highlight it and click the copy button to the right of the add and delete buttons.

You can also add users to a category manually via the second tab on the right hand pane. These users will be included in the category in addition to any rules on the first tab (unless the only rule is to use the manual list). This is very useful for testing, or for creating a category for library staff so you can count the number of accesses that are probably demonstrations. This should not be used for long lists of users.

If you're using ActiveDirectory, the 'memberOf' field is likely to be quite useful as it contains all the security and distribution groups a user is a member of - in an education environment it will almost certainly be able to tell the difference between staff and students. You will usually want to use 'contains' as the condition rather than 'matches' because each value the attribute has contains a string that looks something like this:

```
CN=Students,CN=Users,DC=ad,DC=example,DC=com
```

You can use any fields that allow you to segment your users though - e.g. sometimes different types of users have different email domains, or perhaps only staff have anything in the job-title field.

Anything to watch out for?

To make the best use of the information available to you, you should talk to you IT team about what data is in which fields and whether it's accurate enough for what you might use it for. They may have some interesting ideas about how to categorise things too.

Because 'memberOf' is what is called a multi-valued attribute, it will usually have several values similar to the above and any one of the values matching the condition will assign the category – because of this, the 'not' match types are best avoided for multi-valued attributes such as 'memberOf'.

Next step

[Setting up attributes](#)